CLAIMS

1. A method for separating data blocks referenced by a writable virtual disk (vdisk) from data blocks referenced only by a backing store of a storage system, the method comprising the steps of:

loading blocks of the writable vdisk from a disk into a memory, the loaded blocks including a writable vdisk indirect block having a plurality of fields, each field storing a valid pointer to a data block or an invalid pointer representing a hole;

loading blocks of the backing store from a disk into the memory, the loaded blocks including a backing store indirect block having a plurality of fields, each backing store indirect block field corresponding to a field of the writable vdisk indirect block, one or more backing store indirect block fields having a pointer to a data block;

searching each field of the writable vdisk indirect block for a hole; and

replacing each field having a hole in the writable vdisk indirect block with a new pointer to the data block referenced by the corresponding backing store indirect block field.

2. The method of claim 1 wherein the step of replacing comprises the step of:

dirtying the data block pointed to by the backing store indirect block to enable write allocation of the dirty data block without altering a data content of the data block.

3. The method of claim 1 wherein the step of replacing further comprises the steps of:

choosing a new pointer for a newly allocated data block containing the unaltered data content;

setting bits in block allocation structures for the newly allocated data block; and

placing the new pointer to the newly allocated data block into the field of the writable vdisk indirect block to replace the hole.

4. The method of claim 3 further comprising the step of:

freeing the dirty data block; and

writing the newly allocated data block to disk.

5. The method of claim 4 further comprising the step of:

releasing an association of the writable vdisk to the backing store to thereby separate the writable vdisk data blocks from the backing store data blocks.

6. The method of claim 1 wherein the pointers contained in the writable vdisk indirect block fields and the backing store indirect block fields comprise logical volume block numbers (VBNs).

7. The method of claim 1 wherein the invalid pointers contained in the writable vdisk indirect block fields comprise a zero logical volume block number (VBN).

8. The method of claim 1 wherein the plurality of fields in the writable vdisk indirect block are a writable vdisk level 1 buffer and the plurality of fields in the backing store indirect block are a backing store level 1 buffer.

9. An apparatus for separating data blocks referenced by a writable virtual disk (vdisk) from data blocks referenced only by a backing store of a storage system, the apparatus, comprising:

a backdoor message handler adapted to load blocks of the writable vdisk and backing store from disk into a memory of the storage system;

a writable vdisk indirect block in the memory having a plurality of fields, each field storing a valid pointer to a data block or an invalid pointer representing a hole;

a backing store indirect block in the memory having a plurality of fields, each backing store indirect block field corresponding to a field of the writable vdisk indirect block, each backing store indirect block field having a pointer to a data block;

a special loading function for searching each field of the writable vdisk indirect block for one or more fields representing a hole; and

a write allocator for replacing each field representing a hole in the writable vdisk indirect block with a new pointer to the data referenced by the corresponding backing store indirect block field.

10. The apparatus of claim 9 wherein the write allocator is further adapted to:

choose a new pointer for a newly allocated data block containing an unaltered data content, set bits in block allocation structures for the newly allocated data block, and place the new pointer to the newly allocated data block into the field of the writable vdisk indirect block to replace the hole.

11. The apparatus of claim 10 wherein the write allocator is further adapted to:

free the dirty data block and write the newly allocated data block to disk.

12. The apparatus of claim 9 wherein the backdoor message handler loads the blocks of the writable vdisk and the blocks of the backing store during periods of reduced processing activity.

13. The apparatus of claim 9 wherein the pointers contained in the writable vdisk indirect block fields and the backing store indirect block fields comprise logical volume block numbers (VBNs).

14. The apparatus of claim 9 wherein the invalid pointers contained in the writable vdisk indirect block fields comprise a zero logical volume block number (VBN).

15. The apparatus of claim 9 wherein the plurality of fields in the writable vdisk indirect block comprises a writable vdisk level 1 buffer and the plurality of fields in the backing store indirect block comprises a backing store level 1 buffer.

16. A method for operating a storage system that services access requests to data stored in data blocks on a storage device, the method comprising;

generating a read-only backing store of an organization of data blocks;

generating a writable image of the organization of data blocks, the writable image including references to the backing store;

separating the backing store and the writable image;

deleting the backing store without interrupting the servicing of the access requests.

17. The method of claim 16 wherein the step of separating further comprises the steps of:

searching a plurality of fields of the writable image for indications to reference the backing store;

replacing each indication with a pointer to a newly allocated data block associated with the writable image.

18. The method of claim 16 wherein the indications to reference the backing store are invalid pointer values.

19. An apparatus for separating data blocks referenced by a writable virtual disk (vdisk) from data blocks referenced only by a backing store of a storage system, comprising:

means for loading blocks of the writable vdisk from a disk into a memory, the loaded blocks including a writable vdisk indirect block having a plurality of fields, each field storing a valid pointer to a data block or an invalid pointer representing a hole;

means for loading blocks of the backing store from a disk into the memory, the loaded blocks including a backing store indirect block having a plurality of fields, each backing store indirect block field corresponding to a field of the writable vdisk indirect block, one or more backing store indirect block fields having a pointer to a data block;

means for searching each field of the writable vdisk indirect block for a hole; and

means for replacing each field having a hole in the writable vdisk indirect block with a new pointer to the data block referenced by the corresponding backing store indirect block field.

20. A computer readable medium, including program instructions executing on a computer, the program instructions including instructions for performing the steps of:

loading blocks of the writable vdisk from a disk into a memory, the loaded blocks including a writable vdisk indirect block having a plurality of fields, each field storing a valid pointer to a data block or an invalid pointer representing a hole;

loading blocks of the backing store from a disk into the memory, the loaded blocks including a backing store indirect block having a plurality of fields, each backing store indirect block field corresponding to a field of the writable vdisk indirect block, one or more backing store indirect block fields having a pointer to a data block;

searching each field of the writable vdisk indirect block for a hole; and

replacing each field having a hole in the writable vdisk indirect block with a new pointer to the data block referenced by the corresponding backing store indirect block field.